

# Spectral Efficient Communications employing 1-Bit Quantization and Oversampling at the Receiver

Tim Hälsig, Lukas Landau, and Gerhard Fettweis

Vodafone Chair Mobile Communications Systems

Technische Universität Dresden, 01062 Dresden, Germany

Email: {tim.haelsig, lukas.landau, fettweis}@tu-dresden.de

**Abstract**—To relax power consumption requirements in multi-gigabit/s communications systems low resolution quantization can be used. Information-theoretic results have shown that systems employing 1-bit quantization and oversampling are a viable option for this. This work investigates such a structure under the influence of additive Gaussian noise and two matched pulse shaping filters. It is described how a BCJR algorithm, based on a finite-state channel assumption, can be used to reconstruct symbols of higher order modulation schemes that allow the transmission of more than one bit per symbol. Furthermore, it is shown how symbol sources that are fitted to the 1-bit constraint can significantly improve the error rate performance of the system.

## I. INTRODUCTION

As data rates of communications systems scale towards multi-gigabit per second, the demands posed to the receiver, especially in terms of sampling rate, grow extensively. Finding suitable analog-to-digital converter (ADC) realizations for these systems becomes a challenge, as high sampling rates and resolutions are either unavailable or accompanied by high power consumptions [1]. One approach to circumvent this problem is to use ADCs with reduced precision (e.g., 1-4 bits) [2], which is based on the fact that they can be designed easier and more efficient than their contemporaries with high precision. The most extreme case of reducing the resolution to one bit has been shown to provide promising information-theoretic results [3]. It was furthermore shown that systems with such a strong quantization constraint can benefit strongly from oversampling [4], [5].

Prior work on this subject mostly considered information-theoretic investigations. In [6] the authors proved that the capacity of a channel, which is lost due to 1-bit quantization, can be partially recovered by oversampling the received signal. It was additionally reported that it is possible to trade conversion resolution versus sampling rate to reduce the loss in capacity. Bit error rate (BER) optimal quantizers were investigated in [7]. The presented approach allows a reduction of the required resolution and power consumption of the ADC, while maintaining performance of the regular uniform quantizer. It was shown in [5] that a system with 1-bit quantization, which transmits quadrature amplitude modulated

(QAM) signals, benefits strongly from oversampling in terms of achievable information rate. The authors also showed that the intuitive level of one bit per symbol can be exceeded by exploiting the noise or intersymbol interference (ISI) generated by the channel. This approach of exploiting the ISI of the channel was extended in [8]. It was shown that this technique can make the full information rate of higher order modulation schemes (e.g., 16-QAM) accessible.

This paper focuses on the implementation of a receiver concept that takes advantage of the ISI and whose performance can be measured in terms of error rates. It is therefore the main step for making the information-theoretic results that have been obtained in prior works accessible in a system design.

The rest of the paper is structured as follows: Section II introduces the considered baseband system model. In Section III it is illustrated how the transmitted symbols can be reconstructed from the received symbols using a finite-state channel representation and a trellis-based algorithm. Section IV shows three different symbol source models and discusses their influence on the system. The numerical performance of different system configurations and their implications is examined in Section V. The final Section VI concludes the paper with a brief summary of the main results.

In this paper bold letters describe vectors, such as  $\mathbf{y}_k$ , and sequences of symbols are at times denoted as  $x_{k-L}^k = x_{k-L}, \dots, x_k$  or  $x^k = x_1, \dots, x_k$ .

## II. SYSTEM MODEL

Assume an equivalent baseband system model of a communications system, as seen in Figure 1. The input symbols  $x_k$  are drawn from a finite set  $\mathbb{X}$ , which is determined by the chosen digital modulation scheme, at a symbol rate of  $1/T_s$ . The two pulse shaping filters  $v(t)$  and  $g(t)$  can be chosen arbitrarily and are introduced to generate the necessary interference. We consider the noise  $n(t)$  to have a Gaussian distributed amplitude and a constant spectral density. To create a fixed value for its variance, both the symbol energy and the pulse energies are normalized to one.

The received signal  $z(t)$  can be written as

$$z(t) = \left( \sum_{k=-\infty}^{\infty} x_k \delta(t - kT_s) * v(t) + n(t) \right) * g(t). \quad (1)$$

It is fed into the ADC where the signal is oversampled  $M$ -fold leading to  $M$  samples per symbol duration  $T_s$ , which are

This work has been supported in part by the German Research Foundation in the framework of the Collaborative Research Center 912 "Highly Adaptive Energy-Efficient Computing" and by the European Social Fund in the framework of the Young Investigators Group "3D Chip-Stack Intracnects".

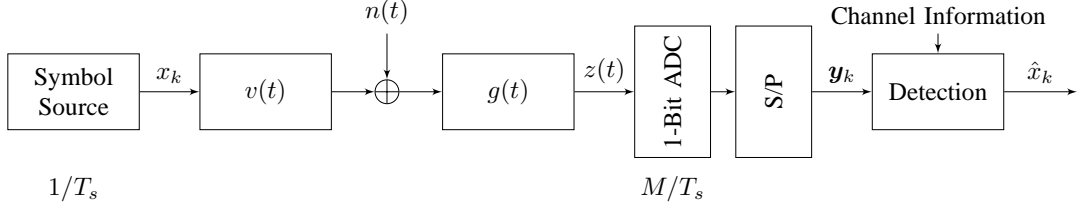


Fig. 1. Equivalent baseband model of a system that transmits data over an AWGN channel with matched filters, and uses oversampling and 1-bit conversion.

stacked in the vector  $\mathbf{y}_k = [y_{k,1}, \dots, y_{k,M}]$ . The samples are furthermore processed by the 1-bit quantization operation  $Q_1$  as

$$y_{k,m} = Q_1 \{z(t)\} = \begin{cases} 1 & \text{for } z(t) \geq 0 \\ -1 & \text{otherwise,} \end{cases} \quad (2)$$

where  $m = 1, \dots, M$  and  $t = kT_s + (m-1)T_s/M$ . The quantized received vectors  $\mathbf{y}_k$  are then fed into the block that tries to recover the input symbols  $x_k$  based on the received symbol sequence  $\mathbf{y}^n$  and channel state information.

We consider the channel to have finite-state (FS) property, where the current output symbol is influenced by the  $L$  previous input symbols. This corresponds to the auxiliary channel assumption

$$P(\mathbf{y}_k | \mathbf{y}^{k-1}, x^k) \approx P(\mathbf{y}_k | x_{k-L}^k). \quad (3)$$

It is assumed that  $L$  is chosen such that the auxiliary channel sufficiently describes the principle property of the channel.

### III. SYMBOL DETECTION WITH BCJR

We seek a method that exploits the FS property of the channel to reconstruct the transmitted symbols. This can be achieved with trellis-based algorithms, like maximum likelihood (ML) or maximum a posteriori (MAP), since the present FS channel can be represented by a trellis. We focus on MAP because it is a generalization of the ML approach which also considers the a posteriori probabilities of the states. The general maximum a posteriori detection rule can be written as

$$\hat{x}_k = \arg \max_{x_k \in \mathbb{X}} P(x_k | \mathbf{y}^n), \quad (4)$$

where  $x_k$  is a symbol at time  $k$ ,  $\mathbb{X}$  is the set of all possible symbols and  $P(x_k | \mathbf{y}^n)$  is the probability for the symbol  $x$  at time  $k$  given the received sequence  $\mathbf{y}^n = \mathbf{y}_1, \dots, \mathbf{y}_n$ .

#### A. BCJR Algorithm

The MAP approach can be efficiently implemented with the BCJR algorithm [9]. From the assumed channel model the states  $s$  at time  $k$  of the trellis are given by

$$s_{k-1} = x_{k-L}, \dots, x_{k-1}. \quad (5)$$

The conventional decomposition of the APP  $P(s_{k-1}, s_k, \mathbf{y}^n)$  leads to the three parts that are required to carry out the

algorithm. First the forward probability  $\alpha$  containing all paths in the trellis leading to a state  $s_k$  can be defined as

$$\begin{aligned} \alpha(s_k) &= P(s_k, \mathbf{y}_1, \dots, \mathbf{y}_k) \\ &= \sum_{\forall s_{k-1}} \alpha(s_{k-1}) \cdot \gamma_k(s_{k-1}, s_k). \end{aligned} \quad (6)$$

With  $\gamma_k$  being the probability for transitioning from state  $s_{k-1}$  to  $s_k$  given by

$$\begin{aligned} \gamma_k(s_{k-1}, s_k) &= P(s_k, \mathbf{y}_k | s_{k-1}) \\ &= P(s_k | s_{k-1}) \cdot P(\mathbf{y}_k | s_k, s_{k-1}). \end{aligned} \quad (7)$$

Furthermore, consider  $\beta$  to be the backward probability as

$$\begin{aligned} \beta(s_{k-1}) &= P(\mathbf{y}_k, \dots, \mathbf{y}_n | s_{k-1}) \\ &= \sum_{\forall s_k} \beta(s_k) \cdot \gamma_k(s_{k-1}, s_k). \end{aligned} \quad (8)$$

It can then be concluded with

$$P(s_{k-1}, s_k, \mathbf{y}^n) = \alpha(s_{k-1}) \cdot \gamma_k(s_{k-1}, s_k) \cdot \beta(s_k), \quad (9)$$

which leads to the APP using

$$P(s_{k-1}, s_k | \mathbf{y}^n) = P(s_{k-1}, s_k, \mathbf{y}^n) / P(\mathbf{y}^n). \quad (10)$$

Since  $P(\mathbf{y}^n)$  is a constant for a given  $\mathbf{y}^n$  the conditional probabilities can be obtained by normalizing the joint probabilities to add up to one. Note that it is necessary to find proper initial values for the recursions of  $\alpha$  and  $\beta$ . Moreover, to efficiently implement the algorithm its matrix form can be used (e.g., [10]).

Applying the state model from equation (5) to the algorithm yields the following equations

$$\begin{aligned} \gamma_k(x_{k-L}^k) &= P(x_k | x_{k-L}^{k-1}) \cdot P(\mathbf{y}_k | x_{k-L}^k) \\ \alpha(x_{k-L+1}^k) &= \sum_{\forall x_{k-L}^{k-1}} \alpha(x_{k-L}^{k-1}) \cdot \gamma_k(x_{k-L}^k) \\ \beta(x_{k-L}^{k-1}) &= \sum_{\forall x_{k-L+1}^k} \beta(x_{k-L+1}^k) \cdot \gamma_k(x_{k-L}^k). \end{aligned}$$

The final a posteriori probability  $P(x_k | \mathbf{y}^n)$ , which is used for detection, can then be computed by summing over the branch APPs  $P(s_{k-1}, s_k | \mathbf{y}^n)$  that correspond to an input symbol of

$x_k$  with

$$\begin{aligned}
P(x_k | \mathbf{y}^n) &= \sum_{\forall s_{k-1}, s_k \supseteq x_k} P(s_{k-1}, s_k | \mathbf{y}^n) \\
&= \sum_{\forall x_{k-L}^k \supseteq x_k} P(x_{k-L}^k | \mathbf{y}^n) \\
&= \sum_{\forall x_{k-L}^k \supseteq x_k} \alpha(x_{k-L}^{k-1}) \cdot \gamma_k(x_{k-L}^k) \cdot \beta(x_{k-L+1}^k).
\end{aligned} \tag{11}$$

### B. Transition Probability

For the algorithm to work it is necessary to identify what the term  $P(x_k | x_{k-L}^{k-1}) \cdot P(\mathbf{y}_k | x_{k-L}^k)$  contains, as it resembles the “Channel Information” that is required prior to starting the algorithm. The first part of this term  $P(x_k | x_{k-L}^{k-1})$  accounts for the probability of the different symbol sequences. To give an example, for independent and uniformly distributed (i.u.d.) input variables this breaks down to  $P(x_k)$ , which is constant and can therefore be omitted when carrying out the algorithm. However, this has to be reassessed when the properties of the symbol source change.

The term  $P(\mathbf{y}_k | x_{k-L}^k)$  is the probability that a certain output symbol  $\mathbf{y}_k$  is observed given the sequence of input symbols  $x_{k-L}, \dots, x_k$ . There are two options to obtain this probability.

If there is knowledge about the appearance of the pulse shaping filter and the properties of the noise, it can be calculated by using a discrete representation of the system. This is done by integrating a multivariate Gaussian distribution over the appropriate quantization intervals. The distribution is characterized by the mean vector  $\boldsymbol{\mu}_k$  given by

$$\boldsymbol{\mu}_k = \mathbf{H} \mathbf{U} x_{k-L}^k, \tag{12}$$

where  $\mathbf{H}$  is the discrete filter matrix with Toeplitz structure of the combined pulse shape  $h(t) = v(t) * g(t)$  and  $\mathbf{U}$  is a upsampling matrix necessary for the discrete representation. The variance of the distribution is the noise covariance matrix  $\mathbf{N}$  with

$$\mathbf{N} = \sigma_n^2 \cdot \mathbf{G} \mathbf{G}^H, \tag{13}$$

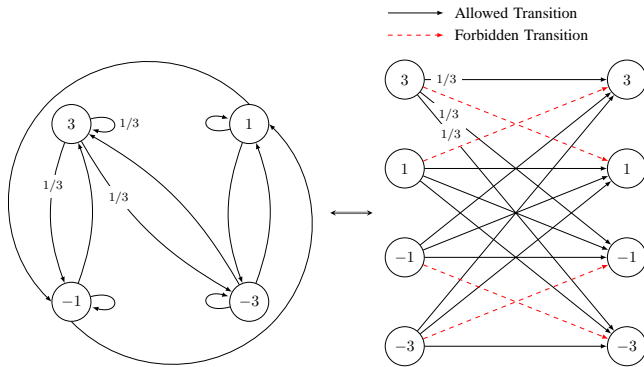


Fig. 2. A Markov source based on 4-ASK, state (left) and trellis (right) representation, avoiding symbol sequences that cannot be uniquely reconstructed. Corresponding transition probabilities are outlined on three example branches.

where  $\mathbf{G}$  is the discrete filter matrix with Toeplitz structure of the receive pulse shaping filter  $g(t)$ .

The other possibility of obtaining  $P(\mathbf{y}_k | x_{k-L}^k)$  is to estimate it. This requires that there is knowledge or at least an assumption about the channel memory  $L$ . A training sequence, which is known in the receiver, can then be sent through the system and in the receiver be used to form the relation between input sequence  $x_{k-L}, \dots, x_k$  and output symbol  $\mathbf{y}_k$ .

### IV. SYMBOL SOURCES

Due to the strong amplitude constraint given by the 1-bit quantizer, different symbol sources shall be considered that can be beneficial for the performance of the system. As the information about the symbol is, when using 1-bit quantization, given by the corresponding zero crossings of the signal, it is assumed that symbol sources generating more zero crossings could help to improve the performance.

#### A. Independent and Uniformly Distributed Source

A commonly used source is the one that draws symbols from a discrete alphabet with an independent and uniform distribution. It has the advantage of having the highest entropy of our considered sources. However, by using this source there is, using 1-bit ADC and BCJR detection, ambiguity when detecting certain sequences of symbols.

To give an example consider symbols drawn from a 4-ASK alphabet, which leads to an entropy of 2 bits for the source. If there is now a sequence of ASK symbols where all of them have the same sign, the sequence does not generate any zero crossings. Therefore, the output sample sequence does not change no matter which of the two possible amplitude levels was actually sent, leading to a possible symbol error.

#### B. Markov State Model

As we have seen in the previous example symbol errors can occur if there is ambiguity when reconstructing the symbols. One solution is therefore to use a source that forbids the transitions which lead to not uniquely identifiable sequences. This can be achieved with a Markov state model, which generally reduces entropy of the source but should improve error

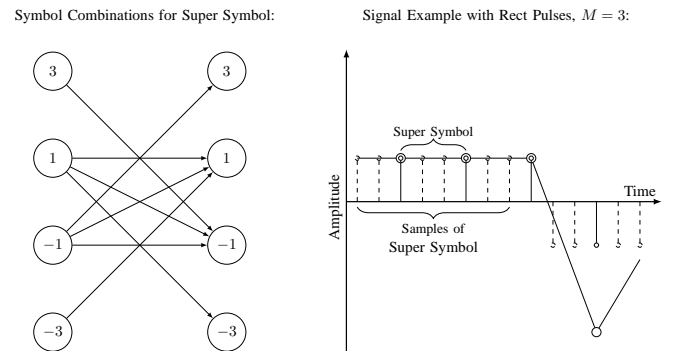


Fig. 3. Possible super symbols constructed from 4-ASK (left) and a signal example with the corresponding sample assignment (right) that produces uniquely identifiable symbols.

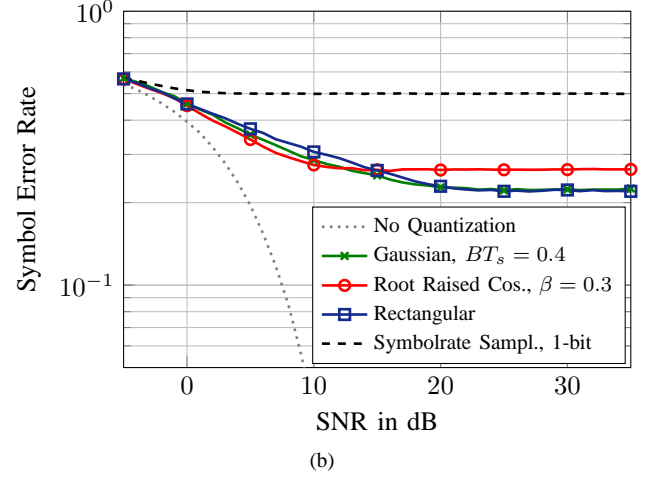
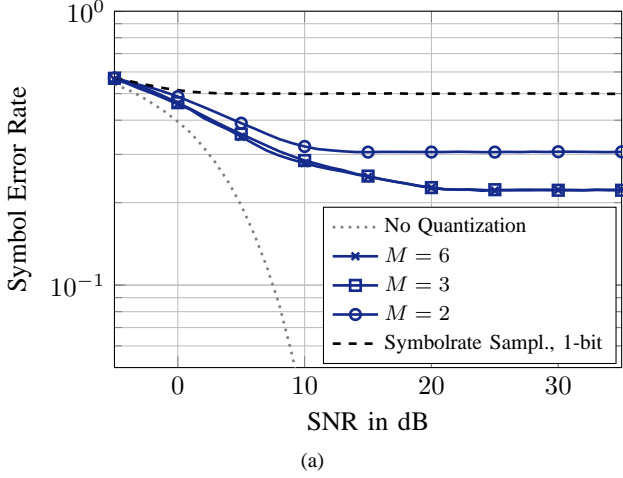


Fig. 4. Symbol error rates for 4-ASK i.u.d. input: (a) Matched rectangular pulses with different oversampling factors  $M$ ; (b) Different pulse shapes (roll-off  $\beta$ , 3-dB bandwidth-symbol time  $BT_s$ ) with oversampling of  $M = 3$ .

rate performance. Another thing that needs to be considered when using this source is the probability  $P(x_k|x_{k-L}^{k-1})$  used in the algorithm. Due to Markov property  $P(x_k|x_{k-L}^{k-1}) = P(x_k|x_{k-1})$ , which is either constant or zero and needs to be adhered to correspondingly in the algorithm.

To give an example, a possible Markov model for 4-ASK and  $M = 3$  can be seen in Figure 2 (state and trellis representation). This source assures that if there are sequences of symbols with the same sign, their values can be uniquely determined based on the right starting symbol. Indeed, there are four symmetric Markov sources yielding the desired property. The one shown here was used as it has the lowest energy in higher frequency components. It is characterized by the states  $x_j$ , which are defined by the previous input symbol, and the state (symbol) transition, which is possible with probability of  $P(x_i|x_j) = 1/3$ , given that the transition is not forbidden. For comparison if we were to include the red dashed lines (trellis representation) and change the transition probability to  $1/4$ , it would yield a conventional 4-ASK i.u.d. source.

The entropy of the source can be calculated by

$$H(X_{\text{Markov}}) = - \sum_{j=1}^4 \sum_{i=1}^4 P(x_j) P(x_i|x_j) \log_2 P(x_i|x_j) \quad (14)$$

$$\approx 1.585 \text{ bit}$$

where  $P(x_j) = 1/4$  is the stationary distribution of the state model. Instead of being able to transmit 2 bits, as was the case for 4-ASK i.u.d., this Markov source can hence only transmit 1.585 bits or more accurately, using  $\log_3$ , 1 trit.

### C. Super Symbols

Another way of ensuring possible error free detection of the symbols, while transmitting more than 1 bit per symbol, is to use a source that transmits certain combinations of symbols, which can be uniquely detected by the receiver structure and shall be called super symbols.

With 4-ASK and  $M = 3$  it is for example possible to construct one super symbol from two regular symbols in a way that there are overall 8 super symbols, which can even be detected in a super-symbol-by-super-symbol manner. One possible realization can be seen in Figure 3. With eight possibilities per two symbols the entropy of the source results to 1.5 bits.

It is possible to carry out a super-symbol-by-super-symbol detection because the sample combination, with oversampling  $M = 3$  and rectangular pulses, is for each super symbol unique. Detection with a BCJR algorithm adjusted to super symbols is viable as well.

## V. NUMERICAL RESULTS

The performance of different system configurations is going to be measured in terms of symbol error rates. For this, a sequence of  $n$  symbols  $x^n$  was transmitted through the system and the difference to the estimated sequence  $\hat{x}^n$  was computed. For the results using the BCJR algorithm, the probability  $P(y_k|x_{k-L}^k)$  was estimated using a random pilot sequence of length  $10^5$ . This length assures, from our experience, that the difference to the actual calculated probability is negligibly small. It is always matched filtering assumed, meaning that  $v(t) = g(t)$ . Furthermore, the channel memory is fixed to one symbol with  $L = 1$ , which still models the channel very good but decreases computational complexity significantly.

Note that we show symbol error rates because going towards a bit level transmission introduces a whole set of additional design tasks (e.g. mapping for the sources), which shall not be part of this work.

### A. 4-ASK i.u.d. Input and BCJR Detection

The results in Figure 4 show the SER behavior of a system with a 4-ASK i.u.d. symbol source. A couple of different conclusions can be drawn from these results. First, with a structure employing 1-bit quantization, oversampling and FS channel assumption, it is possible to achieve a SER of

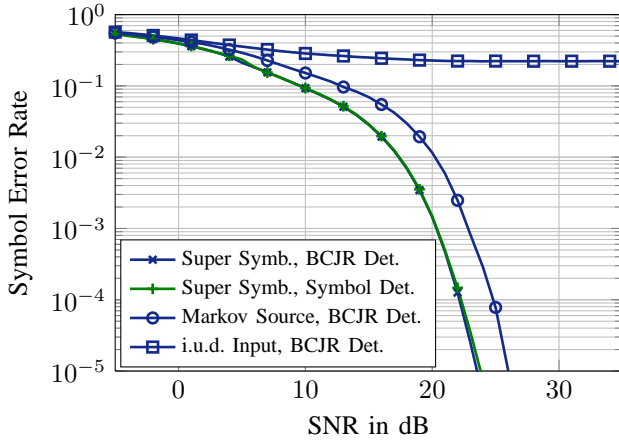


Fig. 5. Symbol error rates for rectangular pulses (duration  $T_s$ ), oversampling  $M = 3$  and different input sources.

0.23 when using 4-ASK. While this is not sufficient for a communications system, it shows that we can reconstruct more than 50% of the symbols.

Secondly, when considering rectangular pulses with duration  $T_s$ , an oversampling increase from  $M = 3$  to  $M = 6$  brings only marginal gain in the lower SNR region. Therefore, an oversampling factor of  $M = 3$  seems to be the most feasible choice for this system structure. It can furthermore be said that the influence of the pulse shape on the performance in the shown cases is rather small. However, the influence can be very strong, depending on the parameters of the pulse (i.e., roll-off, bandwidth). The symbol errors that occur are present due to the fact that there are certain symbol sequences which cannot be uniquely recovered, as discussed earlier.

### B. Markov Source and Super Symbols

When applying the symbol sources from Section IV the error rates improve drastically and indeed tend towards zero, as can be seen in Figure 5. The results were computed with rectangular pulses of duration  $T_s$  and oversampling of  $M = 3$ .

For the Markov source results the model given in Figure 2 was used and the corresponding BCJR detection algorithm was employed. The results look promising but two things have to be kept in mind. First when applying this source to the system the rate is reduced from 2 bits per symbol, which is available with 4-ASK i.u.d., to about 1.58 bits per symbol. Secondly mapping and demapping bits to this Markov source poses some problems regarding efficiency and error sensitivity.

For the super symbol results detection was done with a symbol-by-symbol approach as well as a MAP approach using the BCJR algorithm. For the symbol-by-symbol method the 8 samples of the super symbol (as seen in Figure 3 on the right) were considered and the detected super symbol was chosen to be the one that is most likely to have produced this set of samples. It is also sufficient for unique reconstruction to just consider the 4 samples in the center of the super symbol, but the SNR performance turned out to be slightly worse. The BCJR algorithm was used with the same assumptions as in

the previous section but on a super symbol level. There are numerous possibilities when choosing the two symbols that constitute one super symbol, but the one shown in Figure 3 proved to have the best SNR performance.

The results show that the performance gain from using the more elaborate BCJR detection algorithm compared to the symbol-by-symbol approach is very small. This can be explained by the fact that the interference between two super symbols is, when using rectangular pulses, also very small. As for the Markov source, the entropy is reduced for this source as well, but mapping and demapping can be done very easily by assigning bit blocks to the super symbols.

Similar results can be computed with the two other pulse shapes as well (depending on the pulse parameters).

## VI. CONCLUSION

In this paper a communications system with pulse shaping filters, oversampling and 1-bit quantization has been considered. It was explored how a finite-state channel assumption can be used to design a BCJR algorithm that allows the recovery of more than 50% of 4-ASK symbols with this structure. However, the numerical results reveal that the overall error rate is too high for a practical implementation. Therefore, two alternative symbol sources having efficiencies of more than one bit per symbol were introduced that are more suitable for the given scenario. The results show that both are applicable, with SERs tending towards zero, but come at the cost of reduced rates compared to the standard i.u.d. source. Choosing fitting symbol sources and alphabets is consequently one of the easiest ways to design an efficient communications system with 1-bit quantization constraint.

## REFERENCES

- [1] R. H. Walden, "Analog-to-Digital Converter Survey and Analysis," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 4, pp. 539–550, Apr. 1999.
- [2] J. Singh, S. Ponnuru, and U. Madhow, "Multi-Gigabit Communication: the ADC Bottleneck," in *Proc. IEEE International Conference on Ultra-Wideband*, Sep. 2009, pp. 22–27.
- [3] J. Singh, O. Dabeer, and U. Madhow, "On the Limits of Communication with Low-Precision Analog-to-Digital Conversion at the Receiver," *IEEE Trans. Comm.*, vol. 57, no. 12, pp. 3629–3639, 2009.
- [4] S. Shamai, "Information Rates by Oversampling the Sign of a Bandlimited Process," *IEEE Trans. Inf. Theory*, vol. 40, no. 4, pp. 1230–1236, 1994.
- [5] S. Krone and G. Fettweis, "Communications with 1-Bit Quantization and Oversampling at the Receiver: Benefiting from Inter-Symbol-Interference," in *Proc. IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Sep. 2012, pp. 2408–2413.
- [6] T. Koch and A. Lapidoth, "Increased Capacity per Unit-Cost by Oversampling," in *IEEE 26th Convention of Electrical and Electronics Engineers in Israel*, 2010, pp. 684–688.
- [7] M. Lu, N. Shanbhag, and A. Singer, "BER-Optimal Analog-to-Digital Converters for Communication Links," in *Proc. IEEE International Symposium on Circuits and Systems*, May 2010, pp. 1029–1032.
- [8] L. Landau, S. Krone, and G. Fettweis, "Intersymbol-Interference Design for Maximum Information Rates with 1-Bit Quantization and Oversampling at the Receiver," in *Proc. 9th International ITG Conference on Systems, Communication and Coding*, 2013, pp. 1–6.
- [9] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," *IEEE Trans. Inf. Theory*, vol. 20, no. March 1974, pp. 284–287, 1974.
- [10] M. Tüchler and A. C. Singer, "Turbo Equalization: An Overview," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 920–952, 2011.